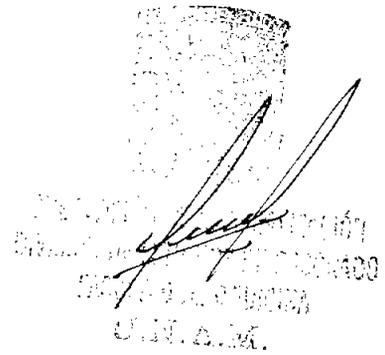


11241

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE MEDICINA  
DIVISIÓN DE ESTUDIOS DE POSGRADO  
DEPARTAMENTO DE PSIQUIATRÍA, PSICOLOGÍA Y SALUD MENTAL

**Modelo computacional de las alucinaciones auditivas en la Esquizofrenia**

*Tesis que, para obtener el título de especialista en psiquiatría, presenta:  
Francisco Camilo de la Fuente Sandoval*



Tutor Teórico  
Dr. Héctor Pérez-Rincón

Tutor Metodológico  
Dr. Luis Xavier Sandoval

2003



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo excepcional.

NOMBRE Francisco Curito de la Fuente Sandoval

FECHA: 31 mayo 2003

FIRMA: 

ESTA TESIS  
LA BIBLIOTECA

## Antecedentes

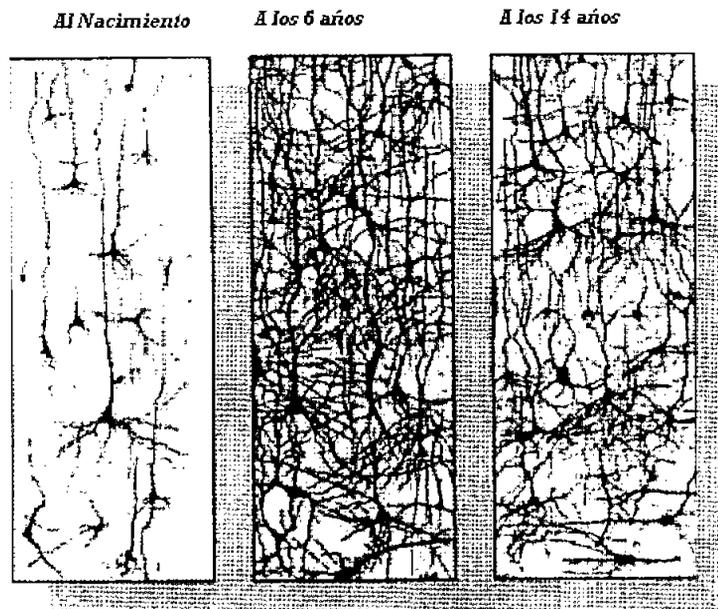
Los modelos computacionales son muy importantes en todas las disciplinas científicas. Su utilidad radica en la simulación de sistemas interactivos complejos. No es la finalidad el reproducir el sistema en su totalidad, sino incorporar ciertas propiedades del sistema estudiado de una manera simplificada de manera que, cuando son simuladas por una computadora, muestran características o conductas que antes no habían sido explicadas.

Los sistemas cerebrales están compuestos por un gran número de elementos neuronales interactivos que han sido explorados de manera útil por medio de la simulación computacional [1]. En esta dinámica es como se explica la simulación de una red neuronal para la simulación de la percepción narrativa del lenguaje. Aunque esta simulación sobresimplifica la gran gama de procesos corticales implicados, ha dado nuevos conocimientos acerca del neurodesarrollo normal y de la inducción de la psicosis [2].

Los elementos para construir el modelo son:

1. Las consecuencias de la pérdida neuronal han sido estudiadas. Estudios en animales indican que la pérdida de neuronas es parte del neurodesarrollo [3]. En estudios *postmortem* en humanos de la corteza frontal media [4] se puede concluir que el desarrollo postnatal normal se caracteriza por una sobre elaboración de procesos neuronales seguido por una reducción gradual de la densidad sináptica hasta cerca de un 60% de los niveles máximos. En el ser humano este proceso se completa alrededor de los 2 años en la corteza occipital y en las áreas prefrontales y de asociación termina hasta la adolescencia [5]. En la figura 1 se ve parte de un estudio de Seeman [6] que demuestra estos cambios:

Figura 1



Esta sinaptogénesis se realiza de manera más o menos aleatoria durante el desarrollo con una subsecuente eliminación selectiva de conexiones más débiles basadas en la experiencia y factores endógenos. Este proceso se conoce como *pruning o podado* [6].

Debido a que la pérdida masiva de neuronas no sucede durante este periodo del desarrollo, la eliminación sináptica de reflejar la reducción en la conectividad entre las neuronas. Menos del 1% de las aferencias a cualquier área cortical derivan del tálamo que es la fuente primaria de entrada no cortical [7]. Por lo tanto, las reducciones de las sinapsis corticales a gran escala deben reflejar reducciones en las conexiones córticocorticales en vez de aferentes talámicas.

2. Muchos investigadores han propuesto que la esquizofrenia es un trastorno del neurodesarrollo [8-11]. La edad característica de inicio de la enfermedad es la adolescencia tardía y la adultez temprana. La pruning prominente durante la adolescencia normal sugiere que la esquizofrenia pudiera surgir como una extensión patológica de este “tardío” proceso del desarrollo [3,8].

Los mecanismos de eliminación neuronal normal y en la Esquizofrenia son desconocidos pero podrían reflejar una sobreactivación (“exitotoxicidad”) o una pobre utilización neuronal [10, 11].

3. La mayoría de los datos que apoyan esto provienen estudios *postmortem* y de neuroimagen, demostrando una densidad medular reducida y disminución en las ramificaciones dendríticas de las células piramidales de la corteza prefrontal en pacientes esquizofrénicos, así como alteraciones en la liberación de glutamato y de ácido  $\gamma$ -aminobutírico. [12,13]

Los estudios sugieren que estos síntomas surgen en el neurocircuito utilizado para el procesamiento perceptual del lenguaje. La percepción es la habilidad de cada sistema de comparar eventos que ocurren simultáneamente en diferentes receptores, un proceso que sirve para traer la máxima respuesta donde la diferencia en la fuerza del estímulo es diferente.

En las últimas fases del procesamiento sensorial, los sistemas hacen comparaciones con eventos pasados y sensaciones recibidas por otros sistemas sensoriales. Estas comparaciones son las bases fundamentales de la percepción, el reconocimiento y la comprensión [14].

La función de cada sistema sensorial es proveer al SNC de una representación del mundo externo.

Debido a los cambios que ocurren al rededor de un individuo, cada sistema sensorial tiene la tarea de traer una representación constantemente actualizada del mundo exterior. Llevar a cabo esta tarea no es fácil ya que requiere una interacción cercana entre el estímulo ascendente y el descendente. Estos dos mecanismos juntos evocan sensaciones, dan lugar a percepciones y activan memorias almacenadas para formar la base de la experiencia consciente. Los mecanismos ascendentes empiezan con la actividad de receptores periféricos, que en conjunto forman una representación neuronal inicial del mundo exterior. Los mecanismos descendentes trabajan para evocar de una gran cantidad de estímulos sensoriales aquellos eventos que requieren de atención inmediata. Al hacer esto los mecanismos descendentes alteran las señales ascendentes de manera de que se optimice la percepción [15].

Algunos estudios sugieren que las alteraciones en el filtrado sensorial o la habituación se asocian con el desarrollo de los síntomas de la esquizofrenia [16]. El filtrado sensorial se conceptualiza como la habilidad para discriminar estímulos triviales. Teóricamente los pacientes con esquizofrenia tienen un defecto en los mecanismos de filtrado que resulta en una sobrecarga sensorial con la consecuente fragmentación cognitiva [17]. Estas anomalías se observan en pacientes con esquizofrenia.

Por otra parte, la habituación representa la forma más simple de aprendizaje y se define como la disminución en la respuesta posterior a la repetición de un estímulo inicial [18]. Teóricamente, las alteraciones en la habituación reflejan la incapacidad para suprimir la

respuesta a un estímulo repetitivo. Los estudios con potenciales evocados auditivos y táctiles sugieren una alteración en la habituación en pacientes esquizofrénicos [19].

Debido a que el lenguaje hablado es una característica propia del ser humano, es sólo el ser humano el único animal en la tierra que padece esquizofrenia [20]. Algunos autores han propuesto que la esquizofrenia es de alguna manera el “precio” que tuvo que pagar el *homo sapiens* en la evolución para acceder al lenguaje [21].

La estrategia de la simulación por computadora no se basa en simular el síndrome completo que caracteriza a la esquizofrenia, sino sólo un síntoma de esta enfermedad, las alucinaciones auditivas, las cuales se reporta que ocurren de 50%-80% de los pacientes con esta enfermedad [22]. Una pista acerca del origen de estas alucinaciones es que comúnmente consisten en lenguaje hablado o “voces” [23], una característica fenomenológica que sugiere que el lenguaje alucinatorio involucra sistemas neuronales dedicados a la percepción del lenguaje auditivo. Esto es apoyado por diversos estudios de imagen como la tomografía por emisión de positrones (PET) y resonancia magnética funcional los cuales evidencian la activación de la corteza de asociación auditiva del lenguaje cuando se presentan las alucinaciones auditivas en los pacientes con esquizofrenia [24, 25]. Por lo tanto, algunos aspectos del sistema de percepción fueron simulados para determinar si el podado de las conexiones del sistema (que simulan a las córticocorticales) podían generar alucinaciones auditivas. El criterio para identificar esto en el sistema fue la producción de percepciones por parte del mismo en ausencia de alguna señal fonética de entrada, es decir una “alucinación”

El proceso involucra a la memoria de trabajo que utiliza las expectativas basadas en palabras previas y frases para discernir nuevos ruidos verbales o fonemas y transformarlos en palabras con significado.

La simulación de la red neuronal se sugirió por la observación de que el lenguaje común, cuando se produce a velocidades normales, tiene una ambigüedad fonética importante debido a la confusión entre la información fonética y los sonidos de fondo subyacentes [26,27]. Por lo tanto, la percepción de una palabra inserta dentro del lenguaje narrativo no solo depende de la información de entrada correspondiente a la palabra por sí misma sino también a palabras previamente percibidas y el conocimiento intrínseco de cómo las palabras son secuenciadas en unidades mayores de lenguajes [28]. La utilización de la expectativa lingüística de la palabra aprendida para filtrar las entradas auditivas de lenguaje refleja una capacidad importante en la memoria de trabajo la cual fue incorporada a la red neuronal. La memoria de trabajo esta implicada en la percepción normal del lenguaje.

Muchos estudios han demostrado las alteraciones en la memoria de trabajo de los pacientes esquizofrénicos [29,30]. Algunos autores [31] han implicado que la alteración involucra las interacciones entre las áreas frontales y mediales temporales, las cuales se conoce están implicadas, en la memoria de trabajo [32]. Debido a esto la red neuronal se enfocará en el componente de la memoria de trabajo para explorar los efectos de la disminución en la conectividad córticocortical.

El sistema computacional está programado para procesar señales de entrada degradadas en palabras identificables. La cantidad de palabras correctamente generadas (identificadas) que entran serán contadas al igual que el número de palabras incorrectamente generadas

(mal identificadas). Las “voces” o alucinaciones serán calificadas cuando las palabras se generen durante periodos de silencio sin entrada de señal.

### Las redes neuronales artificiales

Una neurona con un vector de entrada único  $R$  es que se muestra a continuación. Aquí los elementos individuales de entrada son representados como:

$$P_1, P_2, \dots, P_R$$

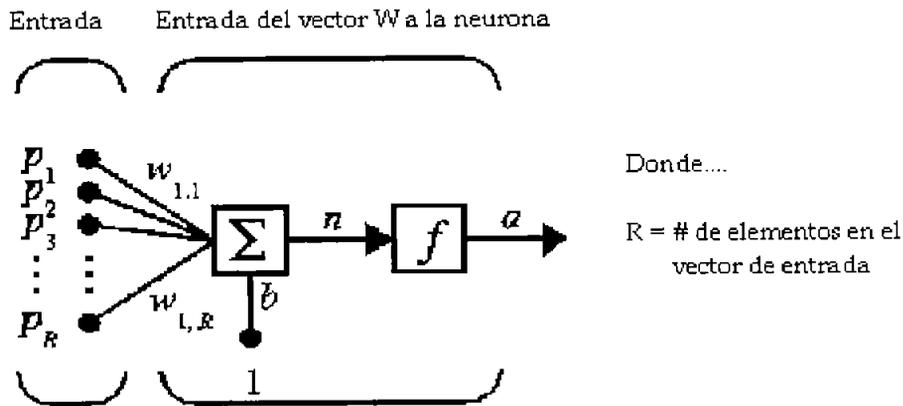
Y que son multiplicados por pesos ( $w$ )

$$W_{1,1}, W_{1,2}, \dots, W_{1,R}$$

Y los valores de peso son alimentados hacia la unión sumatoria ( $\Sigma$ ). La suma se denomina

**Wp.** (figura 2)

Figura 2



La neurona tiene un sesgo  $b$ , el cual es sumado a las entradas sumadas para formar la red de entrada  $n$ . Esta suma,  $n$ , es el argumento para la función de transferencia  $f$ .

$$n = w_{1,1}P_1 + w_{1,2}P_2 + w_{1,2R}P_R + b$$

Esta expresión puede ser escrita en el código del programa MATLAB (35) como

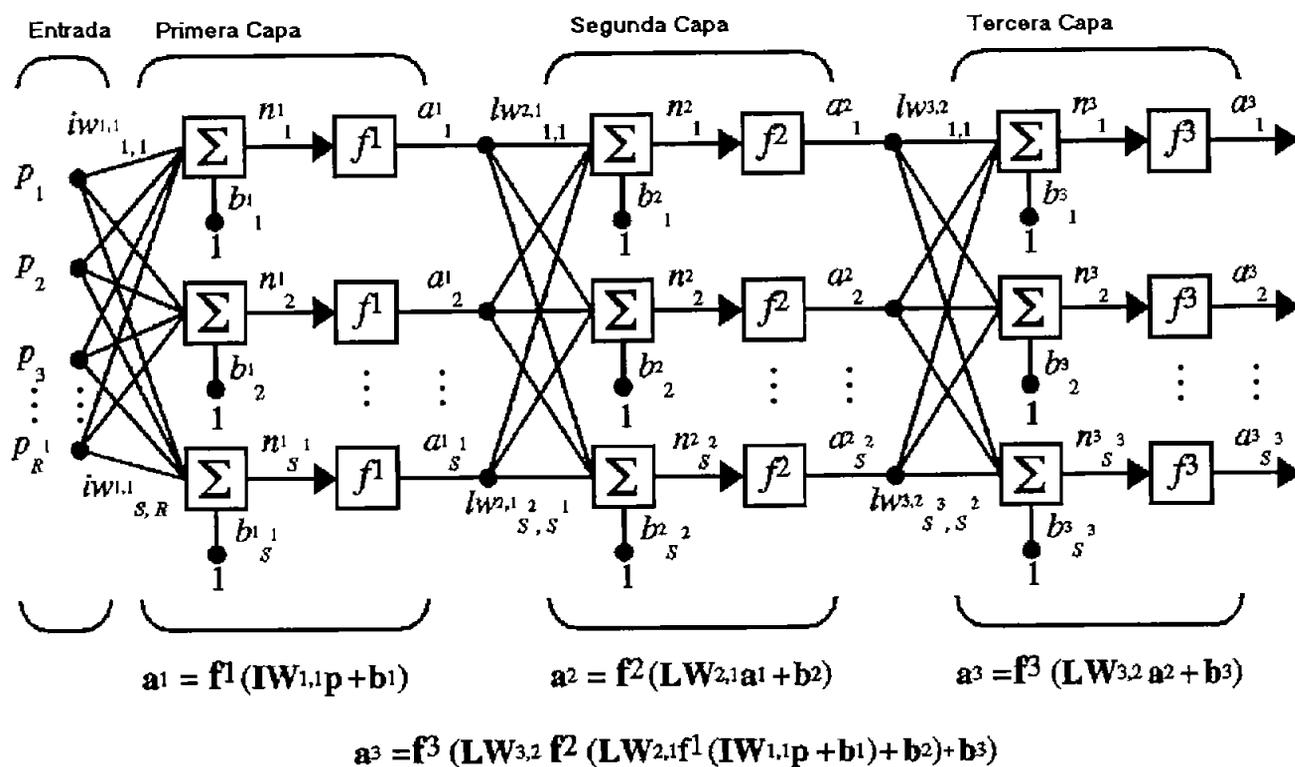
$$n = W * p + b$$

Múltiples capas de neuronas

Una red neuronal puede tener muchas capas. Cada capa tiene un peso matriz  $\mathbf{W}$ , un vector de sesgo  $\mathbf{b}$  y un vector de salida  $\mathbf{a}$ . Para distinguir entre los pesos de las matrices, los vectores de salida, etc. Por cada de estas capas en las figuras, se asignará el número de la

capa como base a la variable de interés. El uso de esta anotación se ve en el esquema inferior con tres capas y la ecuación correspondiente (figura 3):

Figura 3



Esta red tiene entradas  $R^1$ , neuronas  $S^1$  en la primera capa, neuronas  $S^2$  en la segunda capa, etc. Es común que haya diferentes números de neuronas en la diferentes capas. Se alimenta una constante 1 de los sesgos para cada neurona.

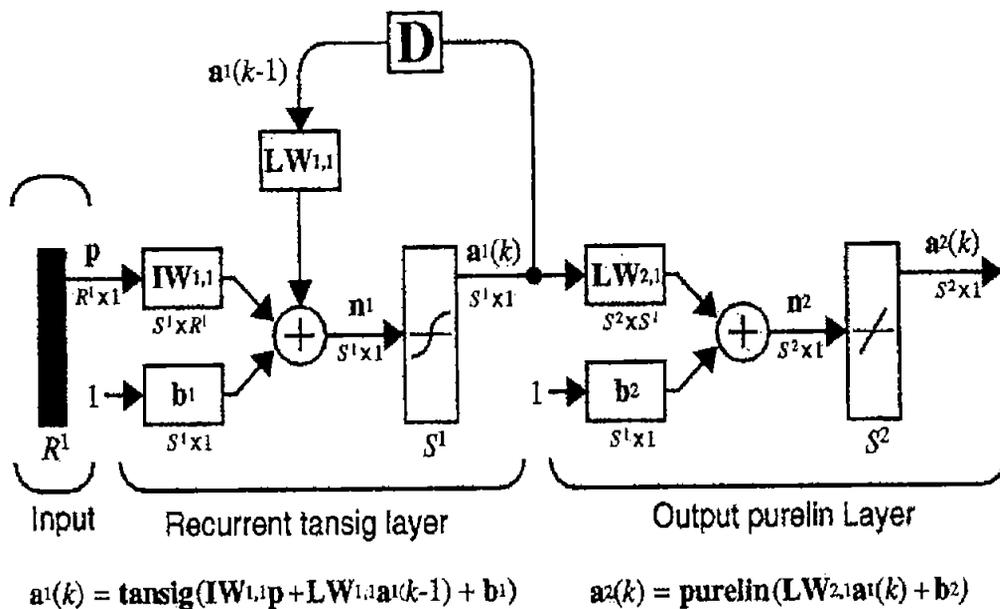
Nótese que las salidas para cada capa intermedia son las entradas de las siguientes capas. Así, la capa 2 puede ser analizada como una capa de red con entradas  $S^1$ , neuronas  $S^2$  y un peso de matriz  $W^2$ ,  $S^1 \times S^2$ . La entrada para la capa 2 es  $a^1$  y la salida es  $a^1$ . Ya que se han

identificado todos los vectores y matrices de la capa 2, puede utilizarse como una sola capa de red por sí misma. Esta aproximación se puede realizar con cualquier capa de la red.

### Modelo de Elman

Las redes neuronales en el modelo de Elman [28,33] comúnmente son redes con dos capas de neuronas con retroalimentación desde la salida de la primera capa a la entrada de la misma capa. Esta conexión recurrente permite a la red de Elman detectar y generar patrones variables en tiempo. A continuación se ilustra en la figura 4 el modelo de Elman.

Figura 4



La red de Elman tiene neuronas TANSIG (función sigmoideal) en la capa oculta (recurrente) y neuronas PURELIN (función lineal) en la capa de salida. Esta combinación es importante en estas redes de dos capas con estas funciones de transferencia.

El único requerimiento es que la capa oculta debe tener un número suficiente de neuronas. Se necesitan un gran número de neuronas ocultas a medida que la función asignada aumenta en complejidad.

Nótese que la red de Elman difiere de las redes de dos capas convencionales ya que la primera capa tiene una conexión recurrente. El retraso en esta capa almacena valores desde el paso en el tiempo anterior, los cuales pueden ser usados en el paso de tiempo presente.

Así, si a dos redes de Elman, con los mismos valores de peso y sesgos, se les dieran entradas idénticas en un momento determinado, sus salidas podrían ser diferentes debido a estados diferentes de retroalimentación [28, 34].

Este modelo utiliza ligas recurrentes entre los elementos que lo componen con el fin de proveer redes que funcionen con una memoria dinámica (simulando a la memoria de trabajo). En este acercamiento las patrones neuronales ocultos reciben una retroalimentación de ellos mismos; las representaciones internas que se desarrollan reflejan las demandas en el contexto de los estados internos previos. Con esto se puede lograr, desde la resolución de problemas muy simples hasta la asignación de características sintácticas y/o semánticas de las palabras. Estas redes son capaces de aprender estas representaciones internas e incorporan las demandas de trabajo con las demandas de memoria lo cual se aproxima a la realidad de que estas dos funciones se encuentran intrínsecamente relacionadas. Estas representaciones son altamente dependientes de contexto mientras también expresan generalizaciones entre las distintos tipos de ítem. Estas representaciones sugieren un método para representar categorías léxicas y la distinción del tipo de palabra [33].

## **Hipótesis**

Si las alucinaciones auditivas se producen, según la teoría propuesta para el pruning, después de la pérdida de más del 60% de las sinápsis, se espera que en el modelo computacional se produzcan percepciones de lenguaje sin ninguna entrada al sistema (lo cual se considerará como alucinaciones auditivas), después de un podaje superior al 60% y que irán aumentando en frecuencia a medida que aumente el podaje. Por otro lado no deberán presentarse estas alucinaciones al eliminar las unidades neuronales ya que esto no estaría de acuerdo con esta teoría propuesta para la esquizofrenia.

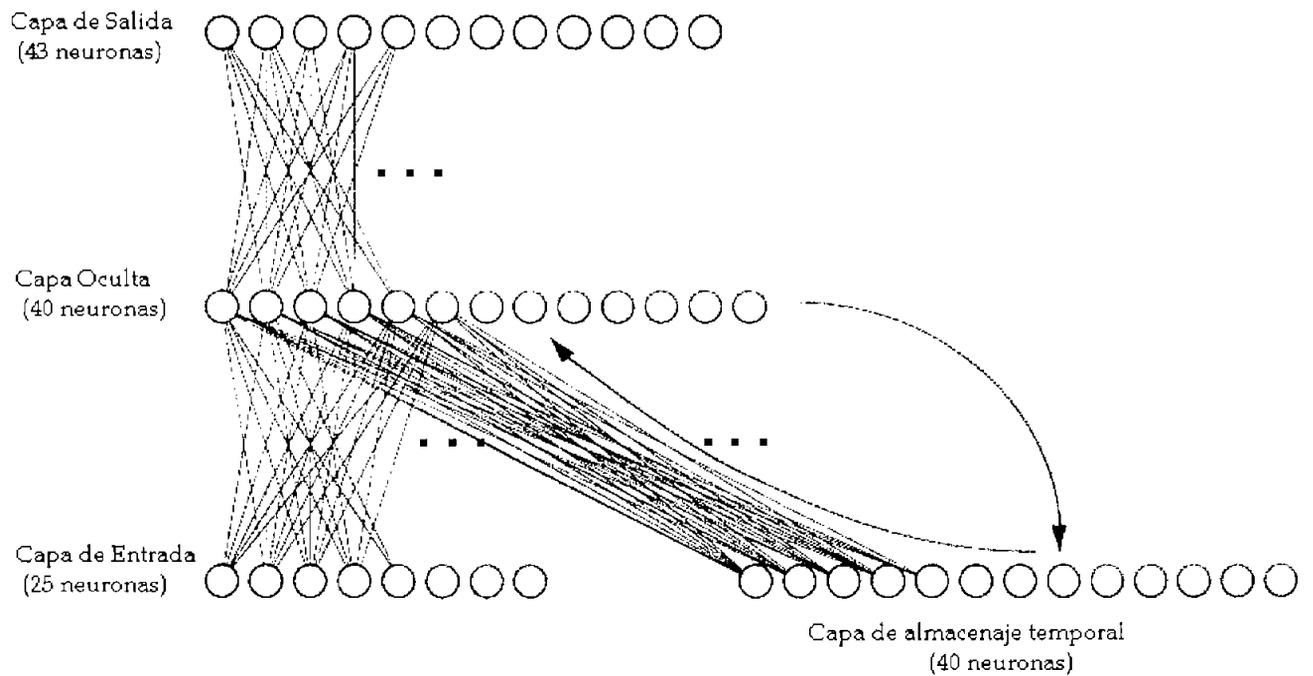
## **Objetivos**

Describir y simular, mediante un modelo computacional, las alucinaciones auditivas provocadas por la disminución en la conectividad sináptica durante el desarrollo, la cual, es una teoría propuesta para el desarrollo de la esquizofrenia.

## **Material y método**

Se utilizó, para simular la red neuronal, una computadora PC con procesador Pentium III a 533 mHz y el programa computacional MATLAB versión 5.3.1.29215<sup>a</sup> [35] con el Toolbox de “Neural Networks” en idioma inglés. Esto se hizo siguiendo el modelo originalmente descrito por Elman [28, 33] para la unidad neuronal y que ha sido utilizado en varias ocasiones para tratar de explicar el proceso de pruning en la esquizofrenia [2, 9, 37].

La red consiste en 148 elementos neuronales divididos en un sistema de cuatro capas (figura 5)



Este sistema se utilizó para traducir las entradas “fonéticas” en palabras. Para esto se asumió que la representación fonética de cada palabra corresponde a un patrón único de activación generado aleatoriamente.

Para alimentar la red se utilizaron palabras asignadas mediante un código binario o de “bits” de 25 combinaciones de los números cero y uno. Se utilizaron palabras en inglés ya que las reglas gramaticales son más sencillas que si utilizara de base el idioma español.

## Patrones de entrada

Para crear este vocabulario primero se realizó la codificación de las frases de manera aleatoria. Como cada palabra tiene una codificación de 25 bits, se creó una matriz de 29 (palabras) x 25 (bits) de manera aleatoria y redondeada a “0” y “1”:

Palabra	Codificación (25 bits)
<i>woman</i>	1 1 1 0 1 0 1 1 1 1 0 0 1 1 1 0 1 1 1 1 0 0 1 1 1
<i>Jane</i>	0 0 1 0 1 1 0 1 1 1 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0
<i>boy</i>	1 1 1 1 1 0 1 0 0 1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 0
<i>girl</i>	0 0 1 0 1 0 1 0 0 1 0 1 1 0 1 0 0 1 1 0 0 1 1 0 1
<i>Bill</i>	1 0 0 0 0 1 0 1 0 0 1 0 1 0 1 0 0 0 1 1 0 0 0 0 1
<i>man</i>	1 1 0 0 0 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 0 0 0 1
<i>cop</i>	0 1 0 1 0 0 0 1 1 1 1 0 0 1 0 0 1 0 0 1 0 1 0 1 1
<i>Sam</i>	0 0 1 1 1 0 0 1 1 1 0 0 0 1 1 1 1 0 0 0 0 1 0 1 1
<i>omen</i>	1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 0 0 1 0 1 1 0 1 1 1
<i>warning</i>	0 1 0 1 1 1 0 0 0 1 1 1 0 0 1 1 1 0 0 0 1 0 1 1 1
<i>story</i>	1 0 1 0 1 0 1 0 1 0 1 1 0 1 1 0 1 1 1 1 0 0 0 1 1
<i>dog</i>	1 1 1 0 0 0 0 0 1 0 0 0 0 1 0 1 1 1 0 1 0 0 1 0 1
<i>God</i>	1 0 0 0 1 1 0 0 0 1 1 0 1 1 0 0 0 1 0 1 0 0 0 0 1
<i>ball</i>	1 1 1 1 0 1 1 0 1 0 0 1 0 0 0 1 1 0 1 0 1 0 1 1 1
<i>chase</i>	0 1 1 1 0 1 0 1 1 0 1 0 0 1 1 0 0 0 0 0 1 1 1 0 0
<i>kiss</i>	0 1 0 1 1 0 0 0 1 1 0 1 1 0 0 1 1 0 1 0 0 0 1 0 1
<i>love</i>	1 0 1 1 1 0 0 1 0 1 1 1 0 1 0 1 1 0 1 0 1 0 1 0 0
<i>fear</i>	1 0 1 0 0 0 0 0 1 1 0 1 1 1 1 0 0 1 1 0 0 0 1 1 0
<i>tell</i>	0 0 1 0 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 1 1 0 1 1 0
<i>run</i>	1 0 1 1 1 0 1 0 0 1 0 1 1 0 0 0 1 1 1 1 0 1 1 0 0
<i>kick</i>	0 1 1 1 0 1 0 0 1 0 0 1 1 1 0 0 1 1 0 0 1 0 1 1 0
<i>give</i>	0 0 1 0 0 1 0 1 0 0 0 1 0 1 1 0 0 1 1 0 1 1 1 1 0
<i>frightens</i>	1 1 0 1 1 1 0 0 1 1 0 0 1 1 0 1 1 1 0 0 1 0 1 1 1
<i>think</i>	0 0 1 1 1 0 1 0 1 0 1 0 1 0 1 0 0 1 1 1 1 0 1 1 0
<i>miss</i>	0 1 0 0 0 1 1 0 0 1 1 0 0 0 1 0 1 1 0 1 1 1 0 0 0
<i>young</i>	0 0 0 1 1 0 0 1 0 0 0 0 0 0 1 1 1 0 1 0 0 0 1 0 1
<i>old</i>	0 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 0 1 0
<i>large</i>	1 1 1 0 0 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 0 1 0 1 1
<i>small</i>	0 1 0 0 0 1 0 1 0 1 1 0 0 0 0 0 1 0 0 0 1 0 1 1 1

El vocabulario creado para la red consistió en 29 palabras, en las cuales se incluían 14 nombres (*woman, Jane, boy, girl, Bill, man, cop, Sam, omen, warning, story, dog, God, ball*), 11 verbos (*chase, kiss, love, fear, tell, run, kick, give, frightens, think, miss*) y 4 adjetivos (*young, old, large, small*). Con este vocabulario se puede realizar una gran variedad de arreglos gramaticales.

Cada una de las 40 neuronas de la capa oculta recibieron la suma de los pesos de las entradas de las 25 neuronas de entrada y las 40 neuronas de almacenamiento temporal que son el equivalente a la memoria de trabajo:

$$\text{Entrada } (x) = \sum_{y=1}^{25} w_{yx} I(y) + \sum_{y=1}^{40} w_{yx} S(y) ,$$

donde la entrada ( $x$ ) es la suma de las entradas recibidas por la neurona  $x$  en la capa oculta,  $I(y)$  es la activación de la neurona  $y$  en la capa de entrada,  $S(y)$  es la activación de la neurona  $y$  en la capa de almacenamiento temporal y  $w_{xy}$  son las proyecciones de los pesos correspondientes (que pueden ser positivos o negativos). La activación de cada neurona de la capa oculta se computó con una función sigmoideal que iba de 0 a 1 y que actuó sobre la suma de las entradas. La capa de salida consistió en 43 neuronas. Las neuronas de la capa de salida recibían entradas exclusivamente de la capa oculta (ver figura 5) y tenían la misma función de activación sigmoideal que las neuronas de la capa oculta.

Además de asignársele un código fonético, a cada palabra se le asignó también un patrón en la capa de salida, en la que de 3 a 6 neuronas se activaban. Estas neuronas codificaban para las características sintácticas y semánticas. Por ejemplo la palabra “*cop*” se representó por

la activación de las neuronas de salida que de manera individual codificaban para SUJETO, ANIMADO y HUMANO, así como una neurona en particular que codificaba así mismo “cop”. En la tabla 1 se presentan algunos códigos de salida para palabras individuales.

Tabla 1

Neuronas de salida (N=43)	Código	Palabras							
		Boy	Jane	Cop	Kiss	Miss	Run	Small	Large
1	Sujeto	●	●	●					
:									
3	Humano	●	●	●					
:									
12	Jane		●						
:									
14	Cop			●					
:									
20	Verbo				●	●	●		
21	Complemento-animado				●	●			
:									
25	Sin Complemento						●		
:									
27	Kiss				●				
28	Miss					●			
:									
39	Adjetivo							●	●
40	Atribución-Edad	●		●					
41	Atribución-Tamaño	●						●	●
42	Diminutivo	●						●	
43	Superlativo			●					●

Cuando la red producía un patrón de activación de la capa de salida, un algoritmo decidía qué palabra correspondía mejor a ese patrón específico. La mejor cabida a esa palabra se

convirtió en “palabra detectada”. Cuando el patrón de activación no encontraba una correspondencia clara la red no detectaba ninguna palabra.

Inicialmente se programó en una rutina de creación de las oraciones, primero comenzando con la codificación de las frases y creando una matriz de manera aleatoria, asignado las correspondencias para los sujetos, verbos, adjetivos, las palabras *won't* y *will* y sus codificaciones individuales.

Una vez definida la codificación para cada palabra, se prosiguió a crear vectores por tipo de palabra, es decir por sujeto, verbo, etc. Después se crearon todas las posibles combinaciones de frases las palabras, resultando poco más de 3400.

Cada frase tiene una estructura: adjetivo + sujeto + verbo + sujeto siguiendo la estructura gramatical del idioma inglés. La programación específica se encuentra en el anexo 1.

Una vez obtenidas las 3400 oraciones se eligieron de manera aleatoria 300 frases definitivas con las cuales se entrenó posteriormente al sistema y que se encuentran en el anexo 2.

Una vez obtenidas las frases se entrenó al sistema utilizando 300 repeticiones, las cuales progresivamente minimizaron el error de los patrones de activación producido por la capa de salida en respuesta a las entradas cuya información fonética fue parcialmente degradada.

Durante el entrenamiento, la red adquirió la habilidad para usar expectativas lingüísticas para guiarse en la detección de palabras (almacenadas como patrones de activación resonando entre las capas oculta y de almacenamiento temporal simulando a la memoria de trabajo).

Comprobación del funcionamiento de la red

Una vez que la red fue entrenada se examinó nuevamente con 300 oraciones diferentes a las utilizadas anteriormente en el entrenamiento, pero en las que se utilizó el mismo vocabulario. Durante esta nueva fase se contaron las frases que identificó y no pudo identificar (cuando confundía una palabra por otra). Para esto se cambiaron dos bits de las entradas diferentes a la utilizadas inicialmente lo cual forzó al sistema en basarse más en la memoria de trabajo y las expectativas lingüísticas para poder detectar correctamente las palabras. Antes del cambio en los 2 bits, de las 300 oraciones se detectaron correctamente 204 que equivalen al 68%. Una vez modificados los bits, se encontró un 62% de eficiencia en la detección correcta de las palabras lo cual nos indica un nivel aceptable de funcionamiento de la red. Cuando el sistema identifica una palabra sin haber tenido una entrada, esto se cuenta como una alucinación.

#### Manipulación de las conexiones (sinapsis) neuronales

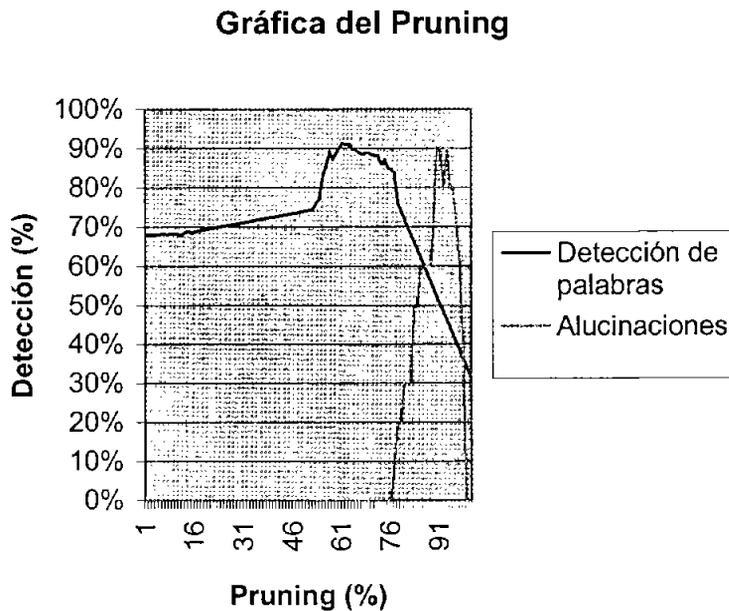
Se simularon dos tipos de manipulación en la estructura de la red neuronal: la primera consistió en la eliminación de las conexiones (pruning) utilizando el concepto Darwiniano, es decir, eliminando la sinapsis según el valor absoluto del peso que tengan y de menor a mayor peso.

La segunda fue la eliminación de las neuronas de manera aleatoria.

## Resultados

El efecto del pruning en las conexiones neuronales se ilustra en la figura 6:

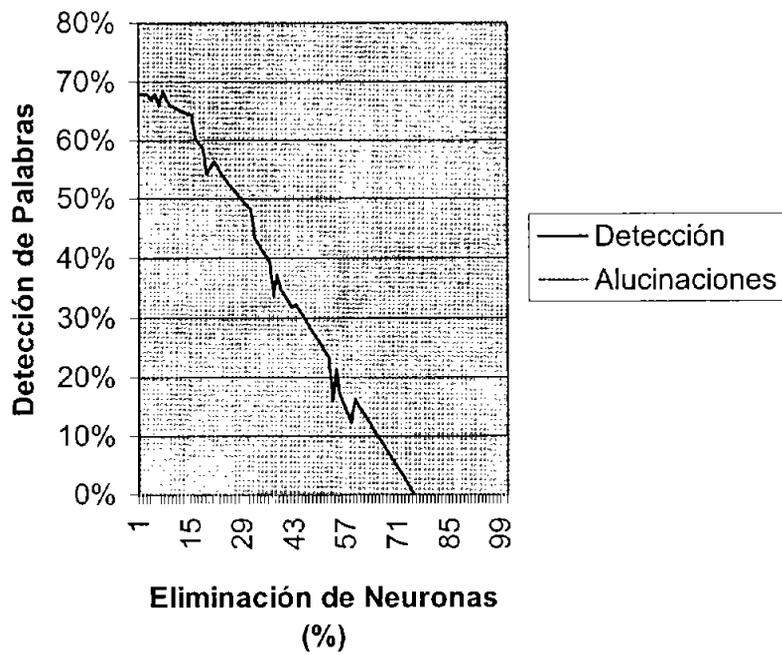
Figura 6



Podemos ver que a medida que se eliminan las sinapsis el porcentaje de detección correcto aumenta del 68% al 91% llegando al la cima cuando hay un pruning aproximado del 60%. Cuando se continuó con la eliminación disminuyó la capacidad de detección, comenzando a presentarse palabras en la ausencia de una entrada, lo cual es considerado como una alucinación. Estas se presentaron con mayor frecuencia cuando el pruning alcanzó entre el 80% y 95%.

Por otro lado, cuando se eliminaron las neuronas en vez de las conexiones, el sistema solo empeoró gradualmente en cuanto a la detección de las palabras y sin presentarse ninguna alucinación (figura 7)

**Gráfica de Eliminación de Neuronas**



## Conclusiones

Como se explicó al principio de este trabajo, los modelos computacionales son muy importantes en todas las disciplinas científicas.

Los sistemas cerebrales están compuestos por un gran número de elementos neuronales interactivos que han sido explorados de manera útil por medio de la simulación computacional. En esta dinámica es como se explica la simulación de una red neuronal para la simulación de la percepción narrativa del lenguaje. Aunque esta simulación sobresimplifica la gran gama de procesos corticales implicados, ha dado nuevos conocimientos acerca del neurodesarrollo normal y de la inducción de la psicosis.

En este trabajo se revisó una simulación por computadora de alucinaciones auditivas de una red neuronal. Los estudios sugieren que estos síntomas surgen en el neurocircuito utilizado para el procesamiento perceptual del lenguaje. La percepción normal del lenguaje es una tarea compleja dada por el alto nivel de ambigüedad acústica del lenguaje cotidiano producido a niveles normales en ambientes ruidosos. El proceso involucra a la capacidad de memoria de trabajo que utiliza las expectativas basadas en palabras previas y frases para discernir nuevos ruidos verbales o fonemas y transformarlos en palabras con significado. El sistema computacional está programado para procesar señales de entrada degradadas en palabras identificables. La cantidad de palabras correctamente generadas (identificadas) que entran fueron contadas al igual que el número de palabras incorrectamente generadas (mal identificadas). Las “voces” o alucinaciones fueron calificadas cuando las palabras se generaron durante periodos de silencio sin entrada.

La detección inicial de palabras degradadas mejoró cuando hasta un 60% de las sinapsis fueron eliminadas. Sin embargo, un pruning arriba del 65% produce una disfunción progresiva en la detección de palabras por la red. A niveles mayor de eliminación (pruning > 80%) las alucinaciones fueron simuladas (como percepciones de lenguaje que sucedían en la ausencia de entrada) con una disfunción posterior del procesamiento del lenguaje. Un estudio posterior en humanos demostró un patrón similar de déficits en el procesamiento del lenguaje altamente específico en pacientes esquizofrénicos que presentaban alucinaciones.

Sin embargo cuando las unidades neuronales, en vez de las conexiones, fueron eliminadas, solo se obtuvo un empeoramiento progresivo de la percepción (gráfica 7), lo cual confirma la hipótesis inicialmente propuesta y que la fisiopatología de la esquizofrenia (y posiblemente otros trastornos psicóticos) subyacen en la conexión entre las neuronas y no en las neuronas mismas.

Este modelo simula principalmente 2 procesos neurobiológicos que son potencialmente patogénicos: el primero consiste en la disminución estructural (neuroanatómica) la conectividad de un sistema neuronal grande, complejo y autoorganizado. El segundo es neuromodulatorio, alterando la excitabilidad de todos los elementos neuronales en un modo continuo. Este modelo predice que el foco de la disminución de las sinapsis en la psicogénesis se halla en los sistemas de memoria de trabajo y que esto ocurre en una fase tardía.

En este modelo, neurodegeneración se traduce como pérdida de conexiones neuronales a través de procesos del desarrollo, no así, pérdida de neuronas mediante procesos neurotóxicos, presentándose la sintomatología solo con una reducción por debajo de un nivel crítico.

Se sugiere que esto surge por dos procesos, ambos relacionados con el desarrollo: una limitada densidad sináptica en la infancia determinada por factores genéticos/perinatales y/o una eliminación de las conexiones sinápticas en la adolescencia y etapas tempranas del adulto. Esto se ha articulado como un modelo de la esquizofrenia, primero del neurodesarrollo y luego neurodegenerativo.

El modelo computacional identifica una densidad sináptica disminuida en las áreas prefrontales y en otras áreas de la corteza de asociación como la vía común final de los síntomas y el curso de la esquizofrenia y, tal vez, de otros trastornos psicóticos. El grado de reducción en la densidad sináptica puede determinar la severidad de la fenomenología y su curso. Si los niveles normales máximos de densidad sináptica en un adulto son de un 40%, se asume que una reducción más allá de este 40% puede desencadenar síntomas prodrómicos, una reducción del 50% puede precipitar síntomas psicóticos y una reducción del 60% puede llevar a una psicosis activa crónica requiriendo la institucionalización del paciente.

Debido a que el pruning es una variable constante este modelo enfatiza la fisiopatología del neurodesarrollo. La variabilidad en la densidad sináptica basal está determinada por la genética, el estrés perinatal y otros procesos.

La asociación de este programa con los síntomas de la esquizofrenia se puede establecer con una extrapolación del programa. Si la actividad cerebral genera pensamientos independientes de un sentido de voluntad, el resultado es la alucinación. Si la actividad lingüística es generada independientemente de la integración prefrontal ejecutoria el resultado es la desorganización del lenguaje. Una consecuencia paralela de la conectividad sináptica reducida es la disminución generalizada de las intercomunicaciones cerebrales

que finalmente se expresa como síntomas negativos. Una disminución del 40% de la sinápsis sugiere que el umbral para generar síntomas es relativamente discreto en donde los pequeños cambios en el porcentaje de las conexiones sinápticas funcionales determina la diferencia entre un estado sin síntomas y los síntomas activos.

En el modelo publicado por McGlashan y Hoffman [33], estos pequeños cambios pueden ser manipulados variando la excitabilidad de las unidades neuronales y así imitando cómo los neurolépticos aumentan el umbral para la formación de síntomas psicóticos y el estrés disminuye este umbral ambos a través de modificaciones en la actividad dopaminérgica. El comienzo de la enfermedad ocurre cuando se alcanza una conectividad sináptica críticamente baja. La esquizofrenia de inicio en la edad temprana surge cuando una red sináptica es expuesta a un proceso de *pruning* anormalmente agresivo, en cambio cuando la enfermedad aparece después de los 30 años se debe más al resultado de un *pruning* normal en una persona cuya red sináptica ya estaba cerca del umbral crítico. Esto podría explicar no sólo a la esquizofrenia sino también otro tipo de patologías que comparten síntomas psicóticos, tales como el trastorno psicótico secundario a sustancias o el trastorno depresivo mayor con síntomas psicóticos en los cuales, sin presentar síntomas tan graves como en la esquizofrenia, las personas que lo presentan podrían tener una red neuronal límite para la psicosis pero sin sobrepasar el umbral para el desarrollo de un trastorno psicótico crónico.

Como se ha mencionado anteriormente, este tipo de modelos no pretenden explicar toda la patología, sin embargo nos acercan cada vez más a entender en mayor medida una de las enfermedades más devastadoras que afectan al ser humano.

## Bibliografia

1. Amit DJ, Brunel N, Tsodyks MV: Correlations of cortical Hebbian reverberations: theory versus experiment. *J Neurosci*; 14: 6435-6445, 1994
2. Hoffman RE, McGlashan TH. Synaptic elimination, neurodevelopment, and the mechanism of hallucinated "voices" in Schizophrenia. *Am J Psychiatry*; 154: 1683-1689, 1997
3. Weinberger DR: Implications of normal brain development for the pathogenesis of schizophrenia. *Arch Gen Psychiatry*; 44: 660-669, 1987
4. Huttenlocker PR. Synaptic density in the human frontal cortex- developmental changes and effect of aging. *Brain Res*; 163: 195-205, 1979
5. Huttenlocker P, Dabholkar A. Regional differences in synaptogenesis on human cerebral cortex. *J Comp Neurol*; 387: 167-178, 1997
6. Seeman, P. Brain development X: Pruning during development. *Am J Psychiatry*; 156: 168, 1999

7. Braitenberg V. Cortical architectonics: general and areal, in *Architectonics of the Cerebral Cortex*. Edited by Brazier MAB, Petsche H. New York, Raven Press, pp 443-465, 1978.
8. Feinberg I: Schizophrenia: caused by a fault in programmed synaptic elimination during adolescence? *J Psychiatr Res*; 17: 319-334, 1982/1983
9. Hoffman RE, Dobscha S: Cortical pruning and the development of schizophrenia: a computer model. *Schizophr Bull*; 15: 477-490, 1989
10. Margolis RL, Chuang D-M, Post RM: Programmed cell death: implications for neuropsychiatric disorders. *Biol Psychiatry*; 35: 946-956, 1995
11. Olney JW, Farber NB: Glutamate receptor dysfunction and schizophrenia. *Arch Gen Psychiatry*; 52: 998-1007, 1995
12. Deakin JWF, Slater P, Simpson MDC, et al: Frontal cortical and left temporal glutaminergic deficiency in schizophrenia. *J Neurochem*; 52: 1781-1786, 1989
13. Sherman AD, Davidson AT, Baruah S, et al: Evidencies of glutamatergic deficiency in schizophrenia. *Neurosci Lett*; 121: 77-90, 1991
14. Siegel, DJ. Perception and cognition en *Kaplan & Sadock's Comprehensive Textbook of Psychiatry*. Williams & Wilkins, 6<sup>th</sup> ed; pp 277-290, 1995

15. Kandel ER. Perception en *Essentials of Neural Science and Behavior*. Appleton & Lange, 1st ed, pp 126-153, 1994
16. Braff DL: Information processing and attention dysfunctions in schizophrenia. *Schizophrenia Bulletin*; 19: 233-259, 1993
17. Braff DL, Geyer MA.: Sensorymotor gating and schizophrenia: Human and animal model studies. *Archives of General Psychiatry*; 47: 181-188, 1990
18. Groves PM, Thompson RF. Habituation: A dual process theory. *Psychological Review*; 77: 419-450, 1970
19. Bolino F, Di Michele V, Di Cicco L, et al.. Sensorimotor gating and habituation evoked by electro-cutaneous stimulation. *Biological Psychiatry*; 36: 670-679, 1994
20. Crow TJ: A Darwinian approach to the origins of psychosis. *Br J Psychiatry*; 167: 12-25, 1995; correction, 167: 414
21. Crow, TJ. Precursors of psychosis as pointers to the Homo sapiens-specific mate recognition system of language. *Br J Psychiatry*; 172: 289-90, 1998
22. Sartorius N, Shapiro R, Jablonsky A: The International Pilot Study of Schizophrenia. *Schizophr Bull*; 1: 21-35, 1974

23. Hoffman RE: Verbal hallucinations and language production processes in schizophrenia. *Behavioral and Brain Sciences*; 9: 503-548, 1986
24. Silbersweig DA, Stern E, Frith C, Cahill C, et al: A functional neuroanatomy of hallucinations in schizophrenia. *Nature*; 378: 176-179, 1995
25. David AS, Woodruff PW, Howard R, et al: Auditory hallucinations inhibit exogenous activation of auditory association cortex. *Neuroreport*; 7: 932-936, 1996
26. Warren RM, Warren RP: Auditory illusions and confusions. *Sci Am*; 223: 30-36, 1970
27. Woods WA: HWIM: A speech understanding system on a computer, en *Models of Language Processes*. Edited by Arbib MA, Caplan D, Marshall JC. New York, Academic Press, pp 95-113, 1982
28. Elman JL: Finding structure in time. *Cognitive Sci*; 14: 179-211, 1990
29. Weinberger DR, Berman KF, Suddath R, Torrey EF: Evidence of dysfunction of a prefrontal-limbic network in schizophrenia: a magnetic resonance imaging and regional cerebral blood flow study of discordant monozygotic twins. *Am J Psychiatry*; 149: 890-897, 1992

30. Park S, Holzman PS: Schizophrenics show spatial working memory deficits. *Arch Gen Psychiatry*; 49: 975-982, 1992
31. Gold JM, Carpenter C, Randolph C, Goldberg TE, Weinberger DR: Auditory working memory and Wisconsin Card Sorting Test performance in schizophrenia. *Arch Gen Psychiatry*; 54: 159-165, 1997
32. Goldman-Rakic PS, Friedman HR: The circuitry of working memory revealed by anatomy and metabolic imaging, en *Frontal Lobe Function and Dysfunction*. Edited by Levin HS, Eisenberg HM, Benton AL. New York, Oxford University Press, pp 72-91, 1991
33. Elman JL: Representation and structure in connectionist models, in *Cognitive Models of Speech Processing: Psycholinguistic and Computational Perspectives*. Edited by Altman GTM. Cambridge Mass, MIT Press, pp 345-382, 1990
34. Wasserman, PD. Fundamentals of artificial neural networks, en *Neural Computing: Theory and Practice*. Edited by Van Nostrand Reinhold. New York, ANZA Research Inc, pp 11-26, 1989
35. MATLAB, The language of technical computing. Version 5.3.1.29215<sup>a</sup> (R11.1). Copyright 1984-1999. The Mathworks, Inc.

36. Hoffman RE, Oates E, Hafner RJ, Hustig HH, McGlashan TH: Semantic organization of hallucinated "voices" in schizophrenia. *Am J Psychiatry*; 151: 1229-1230; 1994
37. McGlashan TH, Hoffman RE. Schizophrenia as a disorder of developmentally reduced synaptic connectivity. *Arch Gen Psychiatry*; 57: 637-648, 2000

## Anexo 1

```
%Este programa, "schizo.m", realiza la preparaci3n inicial de todos los
datos del sistema
%Prepara los datos de entrada "frases" y salida "target table" para la
red neuronal.
% Se crearan dos conjuntos de datos; 1) los que seran utilizados para el
entrenamiento de la red
% y 2) los que seran utilizados para la prueba de eficiencia de la red
despues del entrenamiento.
%

%*****
*****

%rutina de creacion de las frases

%*****
*****

%Primero comenzamos con la codificaci3n de las frases

rand('state',0);           %inicializa el estado aleatorio
inicial (ver comando "rand" de Matlab)
correspondmatrix=round(rand(29,25)); %se tienen un total de 29
palabras. Para cada palabra hay una codificaci3n
%de 25 bits. Por ello se crea una
matriz de 29x25 de manera aleatoria y redondeada
%a "0" y "1".

%asignaci3n de correspondencia para "nouns". Para cada noun se define la
codificaci3n extraida desde "correspondmatrix".

correspwoman=correspondmatrix(1,:);
correspjane=correspondmatrix(2,:);
correspboy=correspondmatrix(3,:);
correspgirl=correspondmatrix(4,:);
correspbill=correspondmatrix(5,:);
correspman=correspondmatrix(6,:);
correspcop=correspondmatrix(7,:);
correspsam=correspondmatrix(8,:);
correspomen=correspondmatrix(9,:);
correspwarning=correspondmatrix(10,:);
correspstory=correspondmatrix(11,:);
correspdog=correspondmatrix(12,:);
correspgod=correspondmatrix(13,:);
correspball=correspondmatrix(14,:);

%asignaci3n de correspondencia para "verbs". Lo mismo para cada verb.

correspchase=correspondmatrix(15,:);
correspkiss=correspondmatrix(16,:);
corresplove=correspondmatrix(17,:);
```

```

correspfear=correspondmatrix(18,:);
corresptell=correspondmatrix(19,:);
corresprun=correspondmatrix(20,:);
correspkick=correspondmatrix(21,:);
correspgive=correspondmatrix(22,:);
correspfrightens=correspondmatrix(23,:);
correspthink=correspondmatrix(24,:);
correspmiss=correspondmatrix(25,:);

```

%asignacion de correspondencia para "adjectives". Lo mismo para cada adjective.

```

correspyoung=correspondmatrix(26,:);
correspold=correspondmatrix(27,:);
corresplarge=correspondmatrix(28,:);
correspsmall=correspondmatrix(29,:);

```

%Una vez definida la codificación para cada palabra, se prosigue a crear vectores por tipo de palabra, es decir por noun, verb etc.

%Se dese tener un manejo de los datos tambien en texto (legibles para el ser humano), para ello se llevara un proceso paralelo.

%aquellos vectores con nombre "\*\*\*data" son de palabras textuales.

% y aquellos con nombre "\*\*\*corresp" son de palabras codificadas.

```

nounsdata=['woman ','Jane ','boy ','girl ','Bill ','man
','cop ','Sam ','omen ','warning','story ','dog ','God
','ball '];

```

```

nounsresp=[correspwoman;correspjane;correspboy;correspgirl;correspbill
;correspman;correspcop;correspsam;correspsam;correspomen;correspstory;cor
respdog;correspgod;correspball];

```

```

verbsdata=['chase ','kiss ','love ','fear ','tell
','run ','kick ','give ','frightens','think ','miss
'];

```

```

verbscorresp=[correspchase;correspkiss;corresplove;correspfear;corresptel
l;corresprun;correspkick;correspgive;correspfrightens;correspthink;corres
pmiss];

```

```

adjectivesdata=['young';'old ','large';'small'];

```

```

adjectivescorresp=[correspyoung;correspold;corresplarge;correspsmall];

```

% a continuacion se crearan todas las posibles frases con todas las palabras. Poco mas de 3400.

%cada frase tiene una estructura: adjective + noun + verb + noun

```
r=1;
```

```
for l= 1:4
```

```
    for i= 1:14
```

```
        for j= 1:11
```

```
            %for k= 1:4
```

```
                for m=1:14
```

```
                    sentence(r).adjective1=adjectivesdata(l,:);
```

```

sentencecorresp(r).adjectivel=adjectivescorresp(1, :);

    if
(1==1 | 1==2) & (i==1 | i==2 | i==3 | i==4 | i==5 | i==6 | i==7 | i==8 | i==12)
    sentence(r).noun1=nounsdata(i, :);
    sentencecorresp(r).noun1=nounscorresp(i, :);
    if
(i==1 | i==2 | i==3 | i==4 | i==5 | i==6 | i==7 | i==8) & (j==1 | j==2 | j==3 | j==4 | j==5 | j==6 |
j==7 | j==8 | j==9 | j==10 | j==11)
        sentence(r).verb=verbsdata(j, :);
        sentencecorresp(r).verb=verbscorresp(j, :);
        if
(j==1) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==12 | m==14)
            sentence(r).noun2=nounsdata(m, :);
            sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
        elseif
(j==2) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==12)
            sentence(r).noun2=nounsdata(m, :);
            sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
        elseif
(j==3) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==11 | m==12 | m==13 | m==14)
            sentence(r).noun2=nounsdata(m, :);
            sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
        elseif
(j==4) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==9 | m==10 | m==11 | m==12 | m=
=13 | m==14)
            sentence(r).noun2=nounsdata(m, :);
            sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
        elseif
(j==5) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==9 | m==10 | m==11)
            sentence(r).noun2=nounsdata(m, :);
            sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
        elseif (j==7) & (m==12 | m==14)
            sentence(r).noun2=nounsdata(m, :);
            sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
        elseif (j==8) & (m==9 | m==10 | m==11 | m==12 | m==14)
            sentence(r).noun2=nounsdata(m, :);
            sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
        elseif
(j==9) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==12)
            sentence(r).noun2=nounsdata(m, :);
            sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
        elseif
(j==10) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==9 | m==10 | m==11 | m==12 | m
==13 | m==14)
            sentence(r).noun2=nounsdata(m, :);
            sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;

```

```

elseif
(j==11) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==11 | m==12 | m==13 | m==14)
    sentence(r).noun2=nounsdata(m, :);
    sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;

    else
    end
elseif (i==9 | i==10 | i==11 | i==12) & (j==9)
    sentence(r).verb=verbsdata(j, :);
    sentencecorresp(r).verb=verbscorresp(j, :);
    if
(j==9) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==12)
        sentence(r).noun2=nounsdata(m, :);
        sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
    else
    end

elseif (i==12) & (j==4 | j==6 | j==9 | j==11)
    sentence(r).verb=verbsdata(j, :);
    sentencecorresp(r).verb=verbscorresp(j, :);
    if
(j==4) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==9 | m==10 | m==11 | m==12 | m=
=13 | m==14)
        sentence(r).noun2=nounsdata(m, :);
        sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
    elseif
(j==9) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==12)
        sentence(r).noun2=nounsdata(m, :);
        sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
    elseif
(j==11) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==11 | m==12 | m==13 | m==14)
        sentence(r).noun2=nounsdata(m, :);
        sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
    else
    end

    else
    end

elseif l==3 & (i==9 | i==10 | i==11)
    sentence(r).noun1=nounsdata(i, :);
    sentencecorresp(r).noun1=nounscorresp(i, :);
    if (i==9 | i==10 | i==11 | i==12) & (j==9)
        sentence(r).verb=verbsdata(j, :);
        sentencecorresp(r).verb=verbscorresp(j, :);
    if
(j==9) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==12)
        sentence(r).noun2=nounsdata(m, :);
        sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
    else
    end

```

```

else
end

elseif l==4 & (i==1 | i==3 | i==4 | i==6 | i==7 | i==12 | i==14)
    sentence(r).noun1=nounsdata(i, :);
    sentencecorresp(r).noun1=nounscorresp(i, :);
    if
        (i==1 | i==2 | i==3 | i==4 | i==5 | i==6 | i==7 | i==8) & (j==1 | j==2 | j==3 | j==4 | j==5 | j==6 |
j==7 | j==8 | j==9 | j==10 | j==11)
            sentence(r).verb=verbsdata(j, :);
            sentencecorresp(r).verb=verbscorresp(j, :);
            if
                (j==1) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==12 | m==14)
                    sentence(r).noun2=nounsdata(m, :);
                    sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
                elseif
                    (j==2) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==12)
                        sentence(r).noun2=nounsdata(m, :);
                        sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
                elseif
                    (j==3) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==11 | m==12 | m==13 | m==14)
                        sentence(r).noun2=nounsdata(m, :);
                        sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
                elseif
                    (j==4) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==9 | m==10 | m==11 | m==12 | m=
=13 | m==14)
                        sentence(r).noun2=nounsdata(m, :);
                        sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
                elseif
                    (j==5) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==9 | m==10 | m==11)
                        sentence(r).noun2=nounsdata(m, :);
                        sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
                elseif (j==7) & (m==12 | m==14)
                    sentence(r).noun2=nounsdata(m, :);
                    sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
                elseif (j==8) & (m==9 | m==10 | m==11 | m==12 | m==14)
                    sentence(r).noun2=nounsdata(m, :);
                    sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
                elseif
                    (j==9) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==12)
                        sentence(r).noun2=nounsdata(m, :);
                        sentencecorresp(r).noun2=nounscorresp(m, :); r=
1+r;
                elseif
                    (j==10) & (m==1 | m==2 | m==3 | m==4 | m==5 | m==6 | m==7 | m==8 | m==9 | m==10 | m==11 | m==12 | m
==13 | m==14)
                        sentence(r).noun2=nounsdata(m, :);

```

```

                                sentencecorresp(r).noun2=nounscorresp(m,:);r=
1+r;
                                elseif
(j==11) & (m==1|m==2|m==3|m==4|m==5|m==6|m==7|m==8|m==11|m==12|m==13|m==14)
                                sentence(r).noun2=nounsdata(m,:);
                                sentencecorresp(r).noun2=nounscorresp(m,:);r=
1+r;
                                else
                                end

                                elseif (i==9|i==10|i==11|i==12)&(j==9)
                                sentence(r).verb=verbsdata(j,:);
                                sentencecorresp(r).verb=verbscorresp(j,:);

                                if
(j==9) & (m==1|m==2|m==3|m==4|m==5|m==6|m==7|m==8|m==12)
                                sentence(r).noun2=nounsdata(m,:);
                                sentencecorresp(r).noun2=nounscorresp(m,:);r=
1+r;
                                else
                                end

                                elseif (i==12)&(j==4|j==6|j==9|j==11)
                                sentence(r).verb=verbsdata(j,:);
                                sentencecorresp(r).verb=verbscorresp(j,:);
                                if
(j==4) & (m==1|m==2|m==3|m==4|m==5|m==6|m==7|m==8|m==9|m==10|m==11|m==12|m=
=13|m==14)
                                sentence(r).noun2=nounsdata(m,:);
                                sentencecorresp(r).noun2=nounscorresp(m,:);r=
1+r;
                                elseif
(j==9) & (m==1|m==2|m==3|m==4|m==5|m==6|m==7|m==8|m==12)
                                sentence(r).noun2=nounsdata(m,:);
                                sentencecorresp(r).noun2=nounscorresp(m,:);r=
1+r;
                                elseif
(j==11) & (m==1|m==2|m==3|m==4|m==5|m==6|m==7|m==8|m==11|m==12|m==13|m==14)
                                sentence(r).noun2=nounsdata(m,:);
                                sentencecorresp(r).noun2=nounscorresp(m,:);r= 1+r;
                                else
                                end

                                else
                                end

                                else

                                %end

                                %sentence(r).verb=verbsdata(j,:);
                                %sentencecorresp(r).verb=verbscorresp(j,:);

                                %sentence(r).adjective2=adjectivesdata(k,:);
                                %sentencecorresp(r).adjective2=adjectivescorresp(k,:);

```



```
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0;      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0];
```

```
transtargettable=targettable';      %transpuesta de la tabla (manejo
de datos )
```

%de las 3400 frases vamos a elegir aleatoreamente 300 frases y crearemos una lista de ellas.

```
for i= 1:300
    n=round(2191*rand)+1;
```

```
lista(i, :)= [sentence(n).adjective1, sentence(n).noun1, sentence(n).verb, sentence(n).noun2];
```

```
listacorresp(:, :, i)= [sentencecorresp(n).adjective1; sentencecorresp(n).noun1; sentencecorresp(n).verb; sentencecorresp(n).noun2];
```

```

%de igual manera creamos el equivalente listado de los datos de salida.

    for j=1:4
        for k=1:31
            if listacorresp(j,:,i)==correspondmatrix(k,:)
                targetdata(j,:,i)=transtargettable(k,:);
            end
        end
    end
end
lista

%se hace un manejo de datos para que sean compatibles con los comandos de
Neural Networks Toolbox.

for i=1:300
    listacorresp1(:,:,i)=listacorresp(:,:,i)';
    targetdata1(:,:,i)=targetdata(:,:,i)';
end

temp=reshape(listacorresp1,25,(4*300));
traininginputdata=temp;
%traininginputdata=con2seq(temp); %set de entrenamiento para entradas.

temp2=reshape(targetdata1,42,4*300);
temp3=[temp2(:,2:1200) temp2(:,1)];
trainingtargetdata=temp2;
%trainingtargetdata=con2seq(temp2); %set de entrenamiento para salidas.

%Esta rutina inicializa la red Nueronal tipo Elman para poder ser
entrenada. Ver detalles con help newelm.
%Nuestra red se llamará "schizonet", y una vez entrenada se llamará
"trainedschizonet"

%crear neural network de tipo elman
inputmatrix(1:25,1)=0; %valor minimo dentro de la matriz
inputmatrix(1:25,2)=1; %valor maximo dentro de la matriz

schizonet=newelm(inputmatrix,[40 42],{'logsig' 'logsig'})% definimos
los parametros de la red
schizonet.trainParam.epochs=300;

%entrenar "schizonet"
trainedschizonet=train(schizonet,traininginputdata,trainingtargetdata);

netdeprueba=trainedschizonet;

for i= 1:30
    n=round(2191*rand)+1;

```

```

listaprueba(i,:)=[sentence(n).adjective1,sentence(n).noun1,sentence(n).verb,
sentence(n).noun2];

listacorresp prueba(:,:,i)=[sentencecorresp(n).adjective1;sentencecorresp(
n).noun1;sentencecorresp(n).verb;sentencecorresp(n).noun2];

    for j=1:4
        for k=1:31
            if listacorresp prueba(j,:,i)==correspondmatrix(k,:)
                targetdataprueba(j,:,i)=transtargettable(k,:);
            end
        end
    end

end
for i=1:30
    listacorresp prueba(:,:,i)=listacorresp prueba(:,:,i)';
    targetdata prueba(:,:,i)=targetdataprueba(:,:,i)';
end
listaprueba

inputdataprueba=reshape(listacorresp prueba,25,(4*30));
%traininginputdata=con2seq(temp); %set de entrenamiento para entradas.

outputtargetdataprueba=reshape(targetdata prueba,42,4*30);
%temp3=[temp2(:,2:1200) temp2(:,1)];
%trainingtargetdata=temp2;
%trainingtargetdata=con2seq(temp2); %set de entrenamiento para salidas.

m=2;
correctos=0; %contamos
equivocados=0;
absdeweith=abs(netdeprueba.LW{1,1});
absdeweith1=reshape(absdeweith,1,1600);
ordenascendente= sort(absdeweith1);
W1=netdeprueba.IW{1,1};
for i=1:1000
    for j=1:40
        for k=1:25
            if abs(netdeprueba.IW{1,1}(j,k))==ordenascendente(i)
                netdeprueba.IW{1,1}(j,k)=0;
            else
                end
        end
    end
    y=sim(netdeprueba,temp); %simulamos la red

    salida=round(y);
    for n=1:1200
        a=isequal(salida(:,n),temp2(:,n));
        if a==1
            correctos=correctos+1;
        else equivocados=equivocados+1;
        end
    end
end

```

```

graficacorrectos(m)=correctos;
graficaequivocados(m)=equivocados;
m=m+1;
correctos=0; %contamos
equivocados=0;
i
end

%como su nombre lo indica, este programa realiza pruebas sobre la Red
Neuronal entrenada

netdeprueba=trainedschizonet; %Creamos una nueva red (copia de la red
entrenada)sobre la cual
%se realizaran las pruebas.
%esto se hace debido a que dicha nueva red
tendra modificaciones
%al realizarse las pruebas y es deseable
mantener la red entrenada
%para posibles futuras aplicaciones.

graficacorrectos=0; %inicializamos contadores.
PalabraMala(1)=0;
PalabraMala(2)=0;
PalabraMala(3)=0;
PalabraMala(4)=0;
PalabraMala(5)=0;
PalabraMala(6)=0;
correctos=0;
equivocados=0;

%simulamos la red con los mismos datos con los que fue entrenada
y=sim(netdeprueba,temp);
salida=round(y);
for i=1:1200

    a=isequal(salida(:,i),temp2(:,i)); %comparamos los datos de salida
con los datos deseables
    if a==1
        correctos=correctos+1 %contamos los que son identificados
correctamente
    else equivocados=equivocados+1 %contamos los identificados
equivocadamente
        numerodefrase=floor((i-1)/4)+1;
        palabra=(i-((numerodefrase-1)*4)); %se desea saber que posicion
en la frase ocupan
        %aquellas que fueron
identificadas equivocadamente

        switch palabra %contamos cuantos
errores hay en:
            case 1, PalabraMala(1)=PalabraMala(1)+1; %la primera palabra
            case 2, PalabraMala(2)=PalabraMala(2)+1; %la segunda "

```

```

        case 3, PalabraMala(3)=PalabraMala(3)+1; %la tercera "
        case 4, PalabraMala(4)=PalabraMala(4)+1; %la cuarta "
        %case 5, PalabraMala(5)=PalabraMala(5)+1; %la quinta "
        %case 6, PalabraMala(6)=PalabraMala(6)+1; %la sexta "

        otherwise
            disp('error, no hay palabra'); % por si las moscas...
        end

    end

end

graficacorrectos(1)=correctos % primer dato de la grafica de correctos
graficaequivocados(1)=equivocados %primer dato de la grafica de
equivocados

%Se realizara a continuacion la eliminacion de las sinapsis por el metodo
darwiniano,
%eliminacion de las de menor a mayor peso (valor).
%k=0;
% absdeweith=abs(netdeprueba.LW{1,1}); %sacamos el valor absoluto de
los pesos
% [b,index]=sort(absdeweith); %encontramos la posicion de las
de menor peso
% for i=1:1600
%     ascendiente
%     b=mean(b);
% [k,l]=find(a<1.5);
% tamano=size(k);
% tamano=tamano(1,1)
% for n=1:40
%     for o=1:40
%         netdeprueba.LW{1,1}(index(n,o),o)=0; %eliminamos el peso (la
sinapsis)

%     y=round(sim(netdeprueba,traininginputdata)); %simulamos la red
%     correctos=0; %contamos
%     equivocados=0;
%     salida=round(y);
%     for i=1:1200
%         a=isequal(salida(:,i),temp2(:,i));
%         if a==1
%             correctos=correctos+1;
%         else equivocados=equivocados+1;
%         end
%     end

%     k=k+1;
%     graficacorrectos(k+1)=correctos;
%     graficaequivocados(n+1)=equivocados;
%     end
% end

%plot(graficacorrectos)

```

## Anexo 2

Young man kiss cop  
old woman miss Sam  
young cop frightens Bill  
old cop frightens girl  
small man miss ball  
young Sam frightens girl  
old Bill love girl  
young boy frightens Jane  
young girl think Bill  
young cop love Bill  
old Bill chase ball  
old boy tell Bill  
small cop love God  
young man kiss dog  
old boy fear girl  
old girl fear story  
small woman think woman  
young Bill fear woman  
small man fear girl  
small man give ball  
young woman love God  
small woman kiss Sam  
small cop love woman  
small boy love woman  
small man frightens boy  
young Bill fear ball  
old cop frightens Sam  
small man think woman  
young Sam frightens Bill  
old Bill miss cop  
old man fear warning  
young woman chase Sam  
small cop miss dog  
small man kiss boy  
old Sam kiss girl  
old Jane tell Jane  
old Sam fear woman  
old man love story  
young cop frightens cop  
small girl kiss man  
young boy tell man  
young cop tell cop  
young boy kiss dog  
old woman tell Sam

young Sam fear ball  
small cop kiss boy  
small boy think boy  
young man think Bill  
young woman miss story  
young woman kiss girl  
old Bill tell girl  
old dog miss Bill  
small boy chase man  
old man miss boy  
young man give dog  
young girl love girl  
old cop love boy  
small cop kiss cop  
small boy love cop  
small man think Sam  
young cop miss man  
young cop chase Jane  
old girl give ball  
young girl frightens Jane  
young Bill frightens Sam  
young Bill think dog  
old cop love boy  
young Jane kiss dog  
young man kiss boy  
old cop tell story  
young cop miss Sam  
young cop think God  
old Sam think boy  
small cop fear girl  
young boy miss man  
young Jane tell Sam  
young boy think cop  
young girl frightens Sam  
small man fear Sam  
young girl chase man  
old man fear God  
young cop chase Jane  
young Bill miss man  
old Sam kick ball  
old girl miss Bill  
small cop chase man  
young Sam fear cop  
old Sam fear omen  
small cop kiss woman  
old Bill frightens woman  
small girl frightens Bill

large omen frightens Bill  
old woman kiss girl  
old Sam think ball  
young girl frightens woman  
small cop fear Sam  
young Bill fear Sam  
small woman fear man  
old man fear Bill  
young girl give ball  
young woman frightens man  
young cop fear ball  
small cop frightens dog  
small cop love Sam  
young man kiss woman  
small cop fear God  
old cop think dog  
young Jane love girl  
old man chase man  
old woman fear dog  
young Bill think omen  
young Bill chase dog  
young Jane frightens Sam  
young woman kiss cop  
small woman frightens woman  
young woman fear omen  
small girl frightens woman  
young Sam miss woman  
old Sam think man  
small cop frightens boy  
young girl tell boy  
young girl frightens woman  
young Sam chase Jane  
young woman frightens Jane  
young dog fear woman  
young woman give story  
small woman think boy  
small dog miss ball  
young boy tell Jane  
old man tell story  
young boy miss story  
young cop miss man  
young girl chase Jane  
young Bill miss ball  
old woman tell man  
young girl chase boy  
young man fear Sam  
small man think Jane

old woman fear Sam  
old girl chase Sam  
young boy chase dog  
young woman tell girl  
young girl kick dog  
small boy miss Jane  
small girl frightens boy  
young Bill kiss dog  
small dog frightens girl  
old Sam tell story  
small girl tell girl  
old boy fear man  
old boy think woman  
young cop fear dog  
young Jane fear man  
young man think ball  
young Bill kiss cop  
small man tell man  
young boy think boy  
young woman love God  
young dog miss girl  
old Sam love Sam  
small girl miss boy  
old Bill miss Bill  
young girl tell Bill  
young Sam chase dog  
young man love boy  
young woman kiss dog  
old woman tell cop  
old cop kiss Sam  
young Jane miss boy  
small woman love woman  
small cop frightens man  
old Sam give dog  
small woman tell girl  
young man fear woman  
young Bill fear cop  
young man miss Jane  
old Sam give warning  
small cop think ball  
old man miss Jane  
old girl think dog  
small boy miss dog  
small woman miss cop  
old cop kick ball  
old cop frightens woman  
small boy fear Bill

small cop frightens cop  
old boy tell Sam  
small boy love dog  
old man miss ball  
young cop think dog  
old cop fear story  
young dog fear woman  
small man miss ball  
old boy tell omen  
young boy chase girl  
old cop frightens cop  
old girl kiss man  
young Bill miss Sam  
old Sam think dog  
young Jane tell Sam  
small cop tell story  
young Bill frightens woman  
young girl give warning  
old man miss girl  
young woman chase woman  
young Sam tell Bill  
young cop kiss man  
young Jane chase woman  
young boy kiss woman  
old woman think Sam  
small boy love story  
small girl tell Jane  
young woman tell Bill  
old Sam miss boy  
small boy miss dog  
old Sam miss man  
small cop fear man  
old cop fear woman  
old dog miss woman  
young Sam frightens Sam  
small girl think Bill  
young Sam think girl  
young Jane fear Jane  
old Sam frightens cop  
small cop fear dog  
young girl tell man  
old woman miss Sam  
young boy kiss woman  
old Jane frightens cop  
small girl fear God  
old woman fear omen  
young man frightens Jane

young Sam miss Bill  
old dog miss boy  
old cop love woman  
small cop chase boy  
small boy give ball  
old boy kiss man  
old man frightens dog  
young Jane love God  
old girl think girl  
old Jane think warning  
small girl love ball  
small girl kiss girl  
old boy love Jane  
small woman give warning  
old cop love ball  
young woman chase woman  
young boy miss cop  
old boy frightens cop  
young woman think God  
young man chase dog  
young Sam fear Jane  
small man kiss boy  
young Sam chase Jane  
young man give dog  
old Jane fear Jane  
small boy think dog  
young Bill kiss boy  
old girl chase woman  
old Jane think boy  
young Sam love story  
old woman kiss Bill  
small girl think warning  
small woman chase girl  
small girl think girl  
old Jane think ball  
small woman miss woman  
young girl chase woman  
young Jane tell woman  
old woman chase girl  
young dog miss ball  
old boy tell woman  
small cop frightens Bill  
young Sam frightens woman  
young man frightens Jane  
old Bill frightens Sam  
old girl fear girl  
young girl frightens Jane

old boy tell cop  
old boy think woman  
small boy think God  
small boy chase girl  
small girl kiss dog  
old man love story  
old Bill love God  
old man fear Jane  
young boy love ball  
young girl tell omen  
old woman miss Jane  
old Bill kiss Sam  
young cop chase Jane  
small woman tell omen  
old woman fear woman  
young woman frightens man  
old Bill think woman  
old Bill kiss girl  
young Bill tell girl  
young Jane miss story  
small man miss woman